Overview

TBrun® provides a fully automated unit test solution that enables developers and testers to apply the powerful static code analysis results of the LDRA tool suite to empower the unit level process.

TBrun makes use of the comprehensive control/data flow analysis provided by LDRA Testbed®. The information extracted includes details concerning the unit interface, parameters, globals (input and output), return values, variable types and usage and procedure calls.

Traditionally this level of information could only have been specified by a developer with an expert knowledge of the unit under test. Hence, in automating this process, TBrun frees up highly qualified staff who may then be re-assigned to other modeling, design and development tasks.

TBrun facilitates several test scenarios:
- Single procedures, Functions, Methods (Unit Test)
- Files containing many functions, Classes (Module Test)
- Complete programs (Sub System & System test)

LDRA has revolutionised the traditional “unit testing” activity, which is typically performed on the host and/or target systems, with its new testing capability, eXtreme Testing.

Key Features

The key benefit of LDRA’s unit testing process, provided by TBrun, is the high degree of automation that the LDRA tool suite provides. This saves both time and resources, thereby enabling a quicker time to market.

Unit Testing / Module Testing with TBrun®

TBrun supports the creation of test cases and execution of them in multiple environments, namely:
- Host / Host
- Host / Target
- Host / Simulator

LDRA’s Unit Testing Features:
- Automated test driver / harness generation with no manual scripting requirement
- High levels of test throughput may be achieved via the intuitive graphical and command line interface options
- Sophisticated automated analysis facilities which reduce test effort, freeing up developers and empowering testers
- Test data and test results are stored and maintained for fully automated regression analysis
- Automated detection and documentation of source code changes
- Tool driven test vector generation
- Facilitates execution of tests in host, target and simulator environments
- Automatically generated test case documentation including pass/fail and regression analysis reports
TBrun Features

TBrun enables genuine and full automatic creation of the driver program. The automatically generated driver handles all language features automatically. Some of these scenarios are detailed below:

**eXtreme Testing**
eXtreme Testing is a totally automated solution which ensures that most source code behaviour can be quickly and easily exercised simultaneously recording high levels of code coverage. Consequently, eXtreme Testing automates the unit/module/integration testing processes, including test harness and test vector production and eliminates the traditional time and resource problems associated with bottom-up testing.

**Automatically Generated Driver Program/Test Harness**
TBrun utilises sophisticated control flow and data flow analysis techniques to document the interface to the unit under test in full. This level of information then enables TBrun to automatically generate test drivers removing the need for manual scripting.

There are no limitations to the automatically generated driver. It is pure C/C++ & Ada 83/95 (based on the application code) and can be executed on the host or in the target environment.

**Stub Creation**
Stubs can be generated from a single button click. The stub features include return values, check values and messaging hit counts which can be added via the graphical input negating the need for coding.

Stubs can be used for functions, methods, constructors, system calls, packages, generics etc.

**Structures / Arrays / Unions**
TBrun has the ability to be able to expose structure elements required for testing. By specifying values TBrun is able to populate test data.

**Class Handling**
Automatic detection of class hierarchy, TBrun holds users through testing classes and creating instances. Tests can be written for a base class and be automatically applied to derived classes.
TBrun Features

Automatic Resolution of Templated Types
TBrun allows the full testing and stubbing of template classes. As part of this process the user initially defines the type(s) for the templated arguments when creating an object of a templated class. Then when testing the methods, the template types will automatically be substituted for the required type. Attributes, parameters and return variables that take their declaration types from template arguments can be initialised and tested within a test case. Member templates may be tested in the same fashion.

Exception Handling
Exceptions can be automatically caught and test cases can be passed or failed on their generation.

Pointer Handling
TBrun detects the use of pointers. The automatically generated driver program will enable the capturing of pointer values for testing, as well as input pointer wizards.

Additional Automatically Handled Language Features:
- Abstract Class Testing
- Automatic Creation of Compound
- Objects in Test
- Access to Private Data
- Re-use of Tests through Class Hierarchy
- Polymorphism Handling
- Inheritance Handling
- Template Handling
- Automatic Creation & Object "Re-Use" (Through Attachment)
- Access Methods & Attributes through the Entire Hierarchy
- Exception Handling
- Pointer Handling
- Generics (Ada)
- In / Out Parameters (Ada)
- Records (Ada)
LDRA's Automated Unit Testing Process with TBrun

Create New Tests

1. Analyse source code to create or update fully populated unit test environment. Select the functions / classes / methods / procedures to test.
   Unit / Module Test

2. Implement test strategy or test plan.
   - Requirements based testing
   - Path based testing
   - Combination strategy

3. Populate the test case(s) with data values and optionally expected outcomes. Default and range values can be added manually or through automatically generated vectors.

4. Compile tests into your chosen test execution environment. Tests have the full flexibility offered by your compiler and target environment.

5. Interrogate results to determine correctness and use test verification to analyse the coverage and data values.

6. Regress tests in an automated process which is easily updated through a hand holding procedure.

Run the tests in:
- Host / Host
- Host / Target
- Host / Simulator environment

Ranges, bounds, exceptions, deltas, arithmetic expressions

Safety Critical Standards Conformance
DO-178B Certification

Enterprise Management Reporting
Publish test Artifacts for Certification
**Client Testimonials**

**QinetiQ Vectored-thrust Aircraft Advanced Control (VRAAC) Harrier**

"LDRA is becoming so popular here that once software engineers have used the tools they are insisting they need to have it available at all times."

Gary Sheppard, Senior Software Engineer, QinetiQ’s Avionic Systems Integration Group (ASiC)

**Neptec’s Laser Camera System in use on Space Shuttle**

"The tools reporting facilities for audit purposes are excellent. The team have also been able to dramatically increase the throughput of unit testing and as a result deliver the project on time."

John Schneider, Director of Engineering, Neptec

**F-35 Lightning II**

"LDRA has the ability to work with Limited Target HW which is important in the automotive sector in order to meet the demands for cost reduction and downsizing. We use the LDRA tool suite as a benchmark for other third-party and similar software platform products."

Akihito Iwai, Project Manager, DENSO Japan

"LDRA has proven they will support us in any way to get the job done especially in meeting demanding milestones. They provided outstanding support for several F-35 teammates: Lockheed Martin (Fort Worth), BAE (Wharton), Northrop Grumman (El Segundo), Seaweed, and Honeywell which directly contributed to a successful first flight of the AA-1 aircraft. We continue to work closely with LDRA to develop the needed automated process support to ensure that our software meets program cost, schedule, and quality targets."

John H. Robb, Air Vehicle Software Senior Manager, LMCO

**Lexus LS460 with world’s first motor-driven electric variable valve timing control system developed by Denso**

**TBrun Availability**

**TBrun is available for the following:**

**Languages**
- Ada 83
- Ada 95
- C
- C++

**Host Platforms**
- Windows 9x/NT/2000/XP/Vista
- Unix
- Linux

**Target Processors**
- ARM
- MIPS
- Freescale
- PowerPC
- Infineon
- Renesas
- Intel
- TI

Other languages, host platforms and target processors are available. Please contact LDRA for more information.

www.ldra.com

LDRA Technology Inc. (US)
Lake Amir Office Park
1250 Bayhill Drive Suite # 360
San Bruno CA 94066
Tel: (650) 583 8880

LDRA UK & Worldwide
Portside, Monks Ferry,
Wirral, CH41 5LH
Tel: +44 (0)151 649 9300
e-mail: info@ldra.com

LDRA Technology UK & Ireland

LDRA Technology France